

GeoAI with Python

*A Practical Guide to
Open-Source Geospatial AI*



Qiusheng Wu

GeoAI with Python

A Practical Guide to Open-Source Geospatial AI

Qiusheng Wu
2026

Contents

Preface	1
Introduction	3
Who This Book Is For	3
What This Book Covers	5
Getting the Most Out of This Book	7
Conventions Used in This Book	8
Downloading the Code Examples	9
Video Tutorials	9
Community and Feedback	10
About the Author	10
Licensing and Copyright	10
I: Foundations	13
1. Introduction to GeoAI	15
1.1. Introduction	15
1.2. Learning Objectives	15
1.3. What is GeoAI?	16
1.4. Key Applications of GeoAI	18
1.5. The Python Ecosystem for GeoAI	21
1.6. Core AI Tasks in Geospatial Analysis	24
1.7. From Traditional Methods to Foundation Models	30
1.8. Key Takeaways	31
1.9. Exercises	32
2. Setting Up Your Environment	33
2.1. Introduction	33
2.2. Learning Objectives	33
2.3. Hardware Requirements	33
2.4. Installing NVIDIA Drivers	34
2.5. Installing Python with Miniconda	36
2.6. Creating a Conda Environment	38
2.7. Installing GeoAI and PyTorch with GPU Support	39
2.8. Installing uv as an Alternative Package Manager	40
2.9. Setting Up Jupyter	41
2.10. Installing Visual Studio Code	42
2.11. Verifying Your Setup	43
2.12. Exercises	45
3. Geospatial Data Essentials	47
3.1. Introduction	47
3.2. Learning Objectives	47
3.3. Raster Data	47
3.4. Vector Data	51
3.5. Coordinate Reference Systems	54
3.6. Annotation Formats for Deep Learning	55
3.7. Image Tiling and Chips	56
3.8. Key Takeaways	57
3.9. Exercises	57

II: Data Acquisition and Preparation	59
4. Downloading Remote Sensing Data	61
4.1. Introduction	61
4.2. Learning Objectives	61
4.3. Satellite Imagery Sources	61
4.4. Searching with STAC	63
4.5. Downloading NAIP Imagery	68
4.6. Downloading Sentinel-2 Data	70
4.7. Downloading Landsat Data	71
4.8. Downloading Vantor Open Data	73
4.9. Accessing Vector Data	73
4.10. Organizing Your Data	76
4.11. Key Takeaways	76
4.12. Exercises	77
5. Interactive Mapping and Visualization	79
5.1. Introduction	79
5.2. Learning Objectives	79
5.3. Creating Interactive Maps with Leafmap	80
5.4. Working with Raster Data on Maps	81
5.5. Visualizing Data from Planetary Computer	84
5.6. Working with Vector Data on Maps	87
5.7. Split-Panel Maps for Comparison	89
5.8. Visualizing Model Results	90
5.9. Best Practices for GeoAI Visualization	92
5.10. Key Takeaways	93
5.11. Exercises	93
6. Creating Training Data	95
6.1. Introduction	95
6.2. Learning Objectives	95
6.3. The Training Data Pipeline	95
6.4. Generating Image Chips from a Single Image	96
6.5. Batch Processing Multiple Images	100
6.6. Batch Processing with Raster Masks	104
6.7. Label Quality Considerations	105
6.8. Dataset Organization	105
6.9. Key Takeaways	107
6.10. Exercises	107
III: Core AI Tasks	109
7. Image Recognition	111
7.1. Introduction	111
7.2. Learning Objectives	111
7.3. Understanding Image Classification	111
7.4. Classification Architectures	113
7.5. Preparing Data for Classification	114
7.6. Import Libraries	115
7.7. Download the EuroSAT RGB Dataset	116
7.8. Explore the Dataset	116
7.9. Train a ResNet-50 Classifier	117

7.10.	Plot Training History	118
7.11.	Evaluate on Test Set	119
7.12.	Plot Confusion Matrix	120
7.13.	Visualize Predictions	121
7.14.	Train an EfficientNet-B0 Classifier	122
7.15.	Compare Results	124
7.16.	Publish and Reuse Models	124
7.17.	Key Takeaways	126
7.18.	Exercises	126
8.	<i>Object Detection</i>	129
8.1.	Introduction	129
8.2.	Learning Objectives	129
8.3.	Understanding Object Detection	129
8.4.	Detection Architectures	130
8.5.	Preparing Detection Datasets	133
8.6.	Evaluating Detection Results	134
8.7.	Import Libraries	135
8.8.	Download the NWPU-VHR-10 Dataset	135
8.9.	Explore the Dataset	136
8.10.	Prepare the Dataset	136
8.11.	Visualize Sample Annotations	137
8.12.	Train a Multi-Class Detection Model	137
8.13.	Plot Training Metrics	138
8.14.	Evaluate with COCO Metrics	139
8.15.	Run Inference on Sample Images	139
8.16.	Visualize Detections	140
8.17.	Batch Inference on Multiple Images	141
8.18.	Publish and Reuse Models	142
8.19.	Key Takeaways	144
8.20.	Exercises	145
9.	<i>Semantic Segmentation</i>	147
9.1.	Introduction	147
9.2.	Learning Objectives	147
9.3.	Foundations of Semantic Segmentation	148
9.4.	Import Libraries	150
9.5.	Building Detection from Aerial Imagery	150
9.6.	Surface Water Mapping	156
9.7.	Cloud and Cloud Shadow Detection	178
9.8.	Land Cover Classification	184
9.9.	Publish and Reuse Models	188
9.10.	Key Takeaways	190
9.11.	Exercises	190
10.	<i>Instance Segmentation</i>	193
10.1.	Introduction	193
10.2.	Learning Objectives	193
10.3.	Instance Segmentation vs. Semantic Segmentation	194
10.4.	Mask R-CNN Architecture	194
10.5.	Downloading the FTW Dataset	196

10.6.	Preparing Training Data	198
10.7.	Training a Mask R-CNN Model	198
10.8.	Running Inference	200
10.9.	Post-Processing Predictions	201
10.10.	Extracting Geometric Properties	203
10.11.	Batch Processing	205
10.12.	Key Takeaways	206
10.13.	Exercises	206
11.	<i>Image Translation</i>	209
11.1.	Introduction	209
11.2.	Learning Objectives	209
11.3.	Foundations of Image Translation	210
11.4.	Import Libraries	211
11.5.	Download Sample Data	212
11.6.	Visualize the Input RGB Composite	212
11.7.	Single-Patch Super-Resolution	213
11.8.	Uncertainty Estimation	215
11.9.	Tiled Inference for Larger Regions	216
11.10.	Limitations and Cautions	218
11.11.	Key Takeaways	219
11.12.	Exercises	220
12.	<i>Change Detection</i>	223
12.1.	Introduction	223
12.2.	Learning Objectives	223
12.3.	Understanding Change Detection	223
12.4.	Traditional Change Detection Methods	225
12.5.	Deep Learning for Change Detection	227
12.6.	Import Libraries	228
12.7.	List Available Models	228
12.8.	Setup	229
12.9.	Download Sample Data	229
12.10.	Visualize Input Imagery	229
12.11.	Initialize the ChangeStar Model	229
12.12.	Run Change Detection	230
12.13.	Visualize Results	231
12.14.	Using the Convenience Function	232
12.15.	Comparing Model Variants	233
12.16.	Adjusting the Threshold	234
12.17.	View Saved Outputs	234
12.18.	Key Takeaways	235
12.19.	Exercises	236
13.	<i>Pixel-Level Regression</i>	239
13.1.	Introduction	239
13.2.	Learning Objectives	239
13.3.	Understanding Pixel Regression	239
13.4.	Regression Architectures	240
13.5.	Case Study: NDVI Prediction from Landsat Imagery	241
13.6.	Evaluation Metrics	247

13.7. Key Takeaways	248
13.8. Exercises	248
IV: Foundation Models and Satellite Embeddings	251
14. Segment Anything for Geospatial	253
14.1. Introduction	253
14.2. Learning Objectives	253
14.3. How SAM 3 Works	254
14.4. Setting Up the Environment	254
14.5. Image Segmentation	255
14.6. Point Prompts for Instance Segmentation	259
14.7. Box Prompts for Building Extraction	266
14.8. Batch Segmentation	270
14.9. Tiled Segmentation for Large Images	272
14.10. Interactive Segmentation	275
14.11. Video Segmentation	276
14.12. Key Takeaways	282
14.13. Exercises	282
15. Vision-Language Models	285
15.1. Introduction	285
15.2. Learning Objectives	285
15.3. How Vision-Language Models Work	286
15.4. Setting Up the Environment	286
15.5. Sample Data	287
15.6. Initializing the Moondream Processor	288
15.7. Image Captioning	288
15.8. Visual Question Answering	289
15.9. Object Detection and Point Localization	290
15.10. Interactive GUI	292
15.11. Sliding Window Analysis for Large Rasters	294
15.12. CLIP-Based Segmentation	300
15.13. Practical Applications in Earth Observation	302
15.14. Limitations and Considerations	303
15.15. Key Takeaways	303
15.16. Exercises	304
16. Satellite Embeddings	305
16.1. Introduction	305
16.2. Learning Objectives	305
16.3. What Are Satellite Embeddings?	305
16.4. Setting Up the Environment	306
16.5. Browsing Available Embedding Datasets	306
16.6. Exploring Patch-Based Embeddings	308
16.7. Visualizing Embeddings	310
16.8. Clustering Embeddings	311
16.9. Similarity Search	314
16.10. Training Classifiers on Embeddings	316
16.11. Comparing Embeddings for Change Detection	320
16.12. Loading Datasets with TorchGeo	321
16.13. Working with TESSERA Temporal Embeddings	322

16.14. Exploring AlphaEarth Satellite Embeddings	327
16.15. Key Takeaways	330
16.16. Exercises	331
V: QGIS Plugins	333
17. Setting Up the GeoAI QGIS Plugin	335
17.1. Introduction	335
17.2. Learning Objectives	335
17.3. Plugin Architecture Overview	335
17.4. Installing the QGIS Plugin	336
17.5. Setting Up Dependencies	339
17.6. Setting Up SAM 3 Access	343
17.7. GPU Memory Management	344
17.8. Checking for Plugin Updates	345
17.9. Key Takeaways	345
17.10. Exercises	346
18. Tree Segmentation in QGIS	347
18.1. Introduction	347
18.2. Learning Objectives	347
18.3. Tree Segmentation Panel Overview	347
18.4. Loading Pre-trained Models	348
18.5. Running Predictions	349
18.6. Exporting Results	352
18.7. Practical Considerations	353
18.8. Key Takeaways	354
18.9. Exercises	355
19. Water Segmentation in QGIS	357
19.1. Introduction	357
19.2. Learning Objectives	357
19.3. Water Segmentation Panel Overview	357
19.4. Configuring Input Data	358
19.5. Running Water Segmentation	359
19.6. Output Options	360
19.7. Practical Considerations	361
19.8. Key Takeaways	362
19.9. Exercises	363
20. Vision-Language Models in QGIS	365
20.1. Introduction	365
20.2. Learning Objectives	366
20.3. Loading the Moondream Model	366
20.4. Image Captioning	368
20.5. Visual Question Answering	369
20.6. Object Detection	370
20.7. Point Localization	371
20.8. Tips and Limitations	372
20.9. Key Takeaways	373
20.10. Exercises	374
21. Segment Anything in QGIS	375
21.1. Introduction	375

21.2. Learning Objectives	375
21.3. Sample Data	375
21.4. Loading a SAM Model	376
21.5. Text-Prompted Segmentation	378
21.6. Interactive Segmentation	378
21.7. Batch Processing	380
21.8. Exporting Results	381
21.9. Key Takeaways	383
21.10. Exercises	384
22. Semantic Segmentation in QGIS	385
22.1. Introduction	385
22.2. Learning Objectives	385
22.3. Creating Training Data	385
22.4. Training a Model	388
22.5. Running Inference	391
22.6. Troubleshooting Common Problems	392
22.7. Key Takeaways	392
22.8. Exercises	393
23. Instance Segmentation in QGIS	395
23.1. Introduction	395
23.2. Learning Objectives	395
23.3. Mask R-CNN Architecture	395
23.4. Creating Training Data	396
23.5. Training Mask R-CNN	398
23.6. Running Inference	399
23.7. Key Takeaways	400
23.8. Exercises	401
References	403

Preface

Introduction

We are in the midst of a revolution in Earth observation. Each day, satellites capture a massive volume of data, including multispectral images, radar, LiDAR, and more. While this information holds the keys to understanding climate vulnerability and urbanization, the gap between data collection and real-world application has remained a major hurdle.

This is where GeoAI comes in.

As the intersection of geospatial science and AI, GeoAI is closing this gap at an incredible pace. Deep learning models can now identify every building at scale or pinpoint individual vehicles across vast areas. New foundation models allow these tools to work in unfamiliar environments with minimal human input. Furthermore, vision-language models and autonomous agents are enabling analysts to query imagery and manage workflows using simple commands. The manual digitization tasks that once took months are now being completed in a matter of minutes.

This book, “*GeoAI with Python: A Practical Guide to Open-Source Geospatial AI*,” is your practical guide to this revolution. It is built on a simple premise: the best way to learn GeoAI is by doing it. Rather than dwelling on abstract theory, every chapter puts real tools in your hands, from preparing training datasets and running deep learning models to visualizing and interpreting your results interactively. The code is real, the datasets are real, and the problems mirror what practitioners encounter every day.

The Python ecosystem for GeoAI has matured dramatically. Libraries like PyTorch, torchgeo, and segment-geospatial bring state-of-the-art deep learning to geospatial practitioners, while packages such as [leafmap](#) and [geoai](#) provide high-level interfaces that make complex workflows accessible. Cloud computing platforms and GPU acceleration have lowered the barrier to training and deploying models at scale. Together, these advances mean that a researcher with a laptop can now accomplish what once required a team of specialists and a room full of servers.

Our journey begins with the foundations: understanding geospatial data formats, setting up a deep learning environment, and mastering interactive visualization. We then cover the essential work of acquiring satellite imagery and preparing training datasets. From there, we progress through the core AI tasks that define modern remote sensing: image recognition, object detection, semantic segmentation, instance segmentation, image translation, change detection, and pixel-level regression. We then explore the frontier of foundation models, including the Segment Anything Model (SAM), vision-language models, and satellite embeddings. Finally, a dedicated section on QGIS plugins brings these AI capabilities into a familiar desktop GIS environment, enabling sophisticated workflows without writing code.

Whether you are a GIS professional seeking to integrate AI into your workflows, a data scientist curious about geospatial applications, a researcher pushing the boundaries of Earth observation, or a student embarking on a career at the intersection of geography and machine learning, this book will equip you with the knowledge and skills to turn satellite imagery into insight.

The future of geospatial analysis is intelligent, automated, and accessible. Let us build it together.

Who This Book Is For

This book is designed for anyone who wants to apply artificial intelligence to geospatial data. If you’ve ever stared at a satellite image wondering how to extract buildings, roads, or land cover automatically, or if you’ve trained deep learning models but struggled to apply them to geographic data with projections, coordinates, and massive file sizes, this book is for you.

You'll Find the Most Value If You Are

A GIS Professional ready to move beyond manual digitization and visual interpretation. You're proficient with QGIS or ArcGIS and understand spatial analysis concepts, but you want to harness deep learning to automate feature extraction, classify land cover at scale, or detect changes across time-series imagery.

A Remote Sensing Scientist or Researcher working with satellite or aerial imagery. You understand spectral bands, spatial resolution, and image preprocessing, but you need a practical bridge to modern AI techniques, from training segmentation models to applying foundation models like SAM to your study areas.

A Data Scientist or Machine Learning Engineer with experience in deep learning who wants to apply your skills to geospatial problems. You're comfortable with PyTorch or TensorFlow and understand CNNs and transformers, but you need guidance on the unique challenges of geographic data: coordinate reference systems, tiling large rasters, handling multi-band imagery, and georeferencing model outputs.

A Graduate Student or Early-Career Researcher in geography, environmental science, urban planning, ecology, or a related field. Your research involves spatial data, and you want to incorporate cutting-edge AI methods into your thesis or publications while building skills that are increasingly in demand.

A Software Developer building geospatial applications that require intelligent analysis. You need to integrate AI-powered feature extraction, classification, or change detection into web applications, APIs, or automated pipelines, and you want to understand the full workflow from model training to deployment.

Essential Prerequisites

You should be comfortable with:

- **Python programming:** variables, functions, classes, and importing libraries (expertise in advanced Python is not required)
- **Basic data analysis:** working with tabular data, filtering, aggregating, and plotting
- **Fundamental geospatial concepts:** understanding that data has a location, what raster and vector data are, and basic familiarity with coordinate systems
- **Command line basics:** navigating directories, running scripts, and installing packages

Helpful Background (But Not Required)

- Experience with deep learning frameworks (PyTorch, TensorFlow)
- Familiarity with remote sensing concepts (spectral bands, spatial resolution, image classification)
- Prior exposure to geospatial Python libraries (rasterio, geopandas, leafmap)
- Understanding of machine learning fundamentals (training vs. inference, overfitting, evaluation metrics)

If You're New to Geospatial Python Programming

If you're new to geospatial Python programming, the following book provides an excellent introduction to both foundational GIS concepts and Python programming:

Wu, Q. (2025). *Introduction to GIS Programming: A Practical Python Guide to Open Source Geospatial Tools*. Independently published. PDF edition ISBN 9798993859712; Print edition ISBN 979-8286979455. Available at gispro.gishub.org.

For those interested in spatial data management and SQL-based geospatial analytics, the companion book offers a comprehensive guide:

Wu, Q. (2025). *Spatial Data Management with DuckDB*. Independently published. PDF edition ISBN 979-8993859705; Print edition ISBN 979-8274710572. Available at duckdb.gishub.org.

What This Book Covers

This book offers a structured journey from geospatial fundamentals to production-ready GeoAI applications, equipping you with practical skills through hands-on examples at every step. Each chapter builds on the previous, progressively expanding your ability to apply AI to real-world geospatial problems.

Part I: Foundations (*Chapters 1–3*)

Establish the essential knowledge and tools that underpin all subsequent content:

- **Chapter 1: Introduction to GeoAI** provides a comprehensive overview of the GeoAI landscape: what it is, why it matters, and how deep learning has transformed geospatial analysis. You'll explore the key AI tasks in remote sensing, survey the Python ecosystem for GeoAI, and understand where foundation models and AI agents are taking the field.
- **Chapter 2: Environment Setup** walks you through configuring a complete GeoAI development environment. From installing Python and managing packages with conda to setting up GPU acceleration with CUDA and PyTorch, this chapter ensures you have a solid, reproducible foundation for all the hands-on work that follows.
- **Chapter 3: Geospatial Data Formats** covers the data formats you'll encounter throughout the book, including raster formats like GeoTIFF and Cloud Optimized GeoTIFF (COG), vector formats like GeoJSON and GeoParquet, and specialized formats for deep learning such as COCO and Pascal VOC annotations. You'll learn to read, write, and convert between formats using Python.

By the end of Part I, you'll have a fully configured deep learning environment and a solid understanding of the data formats and concepts needed to work with GeoAI.

Part II: Data Acquisition and Preparation (*Chapters 4–6*)

Master the critical but often underappreciated work of obtaining and preparing geospatial data for AI:

- **Chapter 4: Downloading Remote Sensing Data** teaches you how to programmatically access satellite imagery, elevation data, and other geospatial datasets from sources like Google Earth Engine, Microsoft Planetary Computer, and various open data portals. You'll learn to search, filter, and download data for your study areas efficiently.
- **Chapter 5: Interactive Mapping & Visualization** introduces powerful visualization tools for exploring geospatial data interactively. Using leafmap and other libraries, you'll create interactive maps, overlay satellite imagery, visualize model predictions, and build compelling visual narratives. These skills are essential for both exploratory analysis and communicating results.
- **Chapter 6: Creating Training Data** addresses one of the most important steps in any AI pipeline: preparing high-quality training datasets. You'll learn to create labeled datasets for segmentation, detection, and classification tasks, including annotation strategies and how to tile large satellite images into training chips.

By the end of Part II, you'll be able to acquire satellite imagery from multiple sources, visualize it interactively, and prepare well-structured training datasets ready for deep learning models.

Part III: Core AI Tasks (*Chapters 7–13*)

Dive into the fundamental AI tasks that define modern geospatial analysis:

- **Chapter 7: Image Recognition** introduces deep learning for classifying entire images. You'll learn to build image classification models that identify scene types, land use categories, and geographic features from satellite and aerial imagery.
- **Chapter 8: Object Detection** teaches you to locate and classify individual objects in geospatial imagery. From detecting buildings and vehicles to identifying trees and solar panels, you'll work with architectures like Faster R-CNN and YOLO adapted for remote sensing.
- **Chapter 9: Semantic Segmentation** covers pixel-level classification, assigning a class label to every pixel in an image. You'll train models to delineate features like water bodies, vegetation, and built-up areas from satellite imagery, learning architectures like U-Net and DeepLabV3+ along the way.
- **Chapter 10: Instance Segmentation** combines detection and segmentation, producing precise boundaries for each individual object. You'll learn to distinguish overlapping features, such as individual building footprints in dense urban areas, using models like Mask R-CNN.
- **Chapter 11: Image Translation** applies AI to transform images between domains, including super-resolution, sensor translation, and cloud removal. You'll train models on multispectral satellite imagery and learn to generate higher-resolution outputs from lower-resolution inputs.
- **Chapter 12: Change Detection** tackles the challenge of identifying what has changed between images captured at different times. From urban expansion and deforestation to post-disaster damage assessment, you'll learn both traditional and deep learning approaches to temporal analysis.
- **Chapter 13: Pixel-Level Regression** extends beyond classification to predict continuous values for each pixel, such as canopy height, biomass, soil moisture, or population density. You'll train regression models on satellite imagery and learn evaluation strategies specific to continuous predictions.

By the end of Part III, you'll have hands-on experience with all major GeoAI tasks, understand when to apply each approach, and be able to train and evaluate models for your own geospatial applications.

Part IV: Foundation Models and Satellite Embeddings (*Chapters 14–16*)

Explore the cutting edge of GeoAI, where pre-trained foundation models are redefining what's possible:

- **Chapter 14: Segment Anything for Geospatial** introduces the Segment Anything Model (SAM) and its application to geospatial imagery. Using the [segment-geospatial](#) package, you'll learn to segment satellite images with minimal prompts, extracting buildings, agricultural fields, water bodies, and more without task-specific training.
- **Chapter 15: Vision-Language Models** explores models that bridge visual and textual understanding. You'll learn to query satellite imagery using natural language, generate captions for geospatial scenes, and leverage multimodal models for tasks like visual question answering on remote sensing data.
- **Chapter 16: Satellite Embeddings** explores the rapidly growing ecosystem of pre-computed satellite embedding datasets from foundation models. You'll learn to browse, load, and visualize embeddings from nine datasets including Clay, TESSERA, and Google's AlphaEarth, perform similarity search and clustering, and train lightweight classifiers on embedding vectors.

By the end of Part IV, you'll understand how foundation models are reshaping GeoAI, and you'll be equipped to apply these advanced techniques to your own research and projects.

Part V: QGIS Plugins (*Chapters 17–23*)

Bring GeoAI capabilities into the familiar QGIS desktop GIS environment through the GeoAI plugin:

- **Chapter 17: Setting Up the GeoAI QGIS Plugin** walks through installing and configuring the GeoAI QGIS plugin. You'll use the built-in dependency installer or set up a Pixi environment with PyTorch and CUDA support, install the plugin, and learn to manage GPU memory for running large AI models within QGIS.
- **Chapter 18: Tree Segmentation in QGIS** demonstrates the Tree Segmentation panel for tree crown detection and ecological object detection. You'll load pre-trained models for trees, birds, livestock, nests, and dead trees, configure prediction parameters for single images and large tiles, and export results in vector, raster, and training data formats.
- **Chapter 19: Water Segmentation in QGIS** covers the Water Segmentation panel for automated water body detection using OmniWaterMask. You'll configure band ordering for different sensors, tune patch-based inference parameters, integrate OpenStreetMap data for refinement, and export water masks as raster and vector data.
- **Chapter 20: Vision-Language Models in QGIS** demonstrates the Moondream VLM panel for natural language interaction with geospatial imagery. You'll generate image captions, ask questions about visible features, detect objects with bounding boxes, and locate features with point markers.
- **Chapter 21: Segment Anything in QGIS** demonstrates the Segment Anything panel for interactive and automated segmentation. You'll load SAM 3, segment objects using text prompts, perform interactive segmentation with point and box prompts, process batches of features, and export georeferenced results.
- **Chapter 22: Semantic Segmentation in QGIS** covers the complete training-to-inference workflow for pixel-level classification without writing code. You'll create training datasets, choose from architectures like U-Net, DeepLabV3+, and SegFormer, train custom models, and run inference with vectorization and smoothing.
- **Chapter 23: Instance Segmentation in QGIS** demonstrates instance-level object detection using Mask R-CNN. You'll create instance training data in PASCAL_VOC format, train models to detect and delineate individual objects, and use per-object output for counting, measurement, and spatial analysis.

By the end of Part V, you'll be able to run sophisticated GeoAI workflows directly within QGIS, making AI-powered analysis accessible to GIS practitioners without requiring Python programming expertise.

Cross-Cutting Themes Throughout

- **Hands-On Practice:** Every concept is accompanied by runnable code examples using real geospatial data.
- **Open Source Tools:** All software used in the book is free and open source, ensuring accessibility and reproducibility.
- **Scalable Workflows:** Techniques that work on a laptop and scale to cloud infrastructure.
- **Real-World Applications:** Examples drawn from environmental monitoring, urban analytics, agriculture, and disaster response.
- **Reproducibility:** All code and data are available on GitHub, enabling you to reproduce every result in the book.

Getting the Most Out of This Book

To maximize your learning experience with this book, consider the following recommendations:

Set Up Your Environment Early: Follow Chapter 2 carefully to configure your Python environment with GPU support. Many GeoAI tasks benefit significantly from GPU acceleration, and a properly

configured environment will save you considerable time throughout the book. If you don't have a local GPU, the book provides guidance on using cloud-based GPU platforms like Google Colab.

Follow Along with the Code: This book is designed to be interactive. Don't just read the code. Type it out, run it, and observe the results. Modify parameters, try different datasets, and experiment. Understanding comes through practice, and the hands-on examples are the heart of this book. When something does not work as expected, resist the urge to skip ahead. Debugging is one of the most valuable learning experiences, and the troubleshooting skills you develop will serve you throughout your GeoAI career.

Work Through the Chapters Sequentially: While experienced practitioners may jump to specific topics, the chapters build on each other. Concepts introduced in earlier chapters (environment setup, data formats, visualization techniques) are used throughout the book. If you skip ahead, refer back when needed.

Use Your Own Data: While the book provides datasets for every example, the real learning happens when you apply these techniques to data you care about. Try running the segmentation models on imagery of your study area, or train an object detection model on features relevant to your research.

Embrace the Errors: Training deep learning models involves iteration. Models won't always converge on the first try, predictions won't always be perfect, and GPUs will occasionally run out of memory. These are learning opportunities. The book addresses common pitfalls and debugging strategies, but developing the instinct to diagnose problems comes through experience.

Build a Portfolio Project: As you progress through the book, identify a geospatial problem that interests you and apply the techniques you're learning. A complete project, from data acquisition to model training to visualization, demonstrates your skills far more effectively than any certification.

Stay Current: GeoAI is a rapidly evolving field. New foundation models, architectures, and tools emerge regularly. The book's GitHub repository is updated periodically with new content and examples for continuing your learning journey.

Conventions Used in This Book

This book uses several conventions to help you navigate the content and understand the code examples:

Code Formatting: All Python code appears in monospaced font within code blocks. When code appears within regular text, it is formatted like this . File and directory names, package names, and function names are also formatted in monospaced font.

Code Examples: Most code examples are complete and runnable. They include comments explaining the key concepts and techniques being demonstrated. Here is a typical example using the `leafmap` library:

```
import leafmap

m = leafmap.Map(center=[40, -100], zoom=4)
m.add_basemap("Esri.WorldImagery")
m
```

Package Imports: At the beginning of each chapter, all required imports are listed. Key packages you'll encounter throughout the book include:

- `geoai` - high-level GeoAI workflows (opengeoai.org)
- `leafmap` - interactive geospatial visualization (leafmap.org)

- `samgeo` - SAM for geospatial data (samgeo.gishub.org)
- `torch` / `torchvision` / `torchgeo` - deep learning with PyTorch
- `rasterio` / `geopandas` - reading and writing geospatial data

Figures and Maps: Interactive maps are rendered directly in Jupyter notebooks. In the print edition, static snapshots are shown with captions describing the interactive features available in the online version.

Downloading the Code Examples

All code examples and supplementary materials for this book are freely available on GitHub: <https://github.com/giswqs/GeoAI-Book>

All sample datasets are available at Source Cooperative: <https://source.coop/opengeos/geoai>

To download the materials, you can use one of the following methods:

- **Clone the repository** (recommended if you have Git installed):

```
$ git clone https://github.com/giswqs/GeoAI-Book.git
```

- **Download as ZIP** (if you prefer not to use Git):
 - Visit the GitHub repository page
 - Click the green **Code** button
 - Select **Download ZIP**
 - Extract the files to your preferred location
- **Browse individual files** online through the GitHub interface or the book's website at book.opengeoai.org

The repository is regularly updated with corrections, improvements, and additional examples. Check back periodically for updates, or **star** and **watch** the repository on GitHub to be notified of changes.

If you find errors in the code or have suggestions for improvements, please open an issue or submit a pull request on GitHub. Community contributions help make this resource better for everyone.

Video Tutorials

Complementing the written content, this book is supported by video tutorials that walk through key concepts and provide additional demonstrations:

<https://youtube.com/@giswqs>

The videos are designed to complement, not replace, the written material. They are particularly helpful for:

- Visual learners who benefit from seeing code being written and executed in real time
- Understanding complex workflows like model training and inference through step-by-step walk-throughs
- Seeing how to approach geospatial problems and debug common issues
- Learning tips and best practices for working with large satellite datasets

Additional video tutorials will be added as the book evolves. Subscribe to the channel to stay notified of new content.

Community and Feedback

I welcome feedback, questions, and suggestions from readers. Your input helps improve the book and makes it more useful for the GeoAI community.

For book-related questions and discussions:

- GitHub Issues: <https://github.com/giswqs/GeoAI-Book/issues>
- GitHub Discussions: <https://github.com/giswqs/GeoAI-Book/discussions>

For package-specific questions:

- `geoai`: <https://github.com/opengeos/geoai/issues>
- `leafmap`: <https://github.com/opengeos/leafmap/issues>
- `segment-geospatial`: <https://github.com/opengeos/segment-geospatial/issues>

Types of feedback that are particularly helpful:

- Errors or unclear explanations in the text or code
- Suggestions for additional examples, datasets, or use cases
- Reports of compatibility issues with different operating systems, GPU configurations, or library versions
- Ideas for new topics or emerging GeoAI techniques to cover
- Success stories of how you've applied the techniques from the book to your own work

About the Author

Dr. Qiusheng Wu is an Associate Professor and the Director of Graduate Studies in the Department of Geography & Sustainability at the University of Tennessee, Knoxville. He also serves as an Amazon Scholar. Dr. Wu's research focuses on advancing open-source geospatial analytics through cloud computing and GeoAI, with an emphasis on leveraging big geospatial data and artificial intelligence to study environmental change, particularly surface water and wetland inundation dynamics.

He is the creator and maintainer of several widely used open-source Python packages, including `geemap`¹ for interactive Google Earth Engine visualization, `leafmap`² for versatile geospatial mapping, `segment-geospatial`³ for applying the Segment Anything Model to geospatial data, and `geoai`⁴ for high-level GeoAI workflows. His open-source projects, available through the `Open Geospatial Solutions`⁵ organization on GitHub, have been widely adopted by researchers, educators, and practitioners worldwide.

Dr. Wu's work bridges remote sensing, Earth observation, and artificial intelligence to make large-scale geospatial data more accessible, reproducible, and intelligent. He is passionate about open science and believes that the best tools for understanding our planet should be freely available to everyone.

Licensing and Copyright

This book embraces the principles of open science and open education. To support transparency, learning, and reuse, the **code examples** in this book are released under a [Creative Commons Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

¹<https://geemap.org>

²<https://leafmap.org>

³<https://samgeo.gishub.org>

⁴<https://opengeoai.org>

⁵<https://github.com/opengeos>

[International \(CC BY 4.0\)](#) license. This means you are free to copy, modify, and distribute the code, even for commercial purposes, as long as appropriate credit is given.

Please attribute code usage by citing the book or linking to the GitHub repository:

Wu, Q. (2026). *GeoAI with Python: A Practical Guide to Open-Source Geospatial AI*. Independently published. PDF edition ISBN 979-8993859729; Print edition ISBN 979-8253507414. Available at book.opengeoai.org.

While the code is freely available, the **text, figures, and images** in this book are **copyrighted** © 2026 by the author and may not be reproduced, redistributed, or modified without explicit permission. This includes all written content, custom diagrams, and embedded visualizations unless otherwise noted.

If you wish to reuse or adapt any non-code material from the book (for example, for teaching, presentations, or publications), please contact the author to request permission.

This dual licensing approach helps balance open access to learning materials with the protection of original creative work. Thank you for respecting these terms and supporting the open-source geospatial community.

A Practical Guide to Open-Source Geospatial AI

Artificial intelligence is transforming how we analyze and understand our planet. **GeoAI with Python** is the definitive practical guide to applying AI techniques to geospatial data, from satellite imagery and aerial photographs to vector datasets and Earth observation archives.

Every chapter includes a companion Jupyter notebook with real datasets, so you can follow along and reproduce every result. Whether you're a GIS professional exploring AI or a data scientist working with spatial data for the first time, this book bridges both worlds.

Qiusheng Wu is an Associate Professor at the University of Tennessee and an Amazon Scholar. He is the creator of several widely used open-source geospatial Python packages, including `geemap`, `leafmap`, `segment-geospatial`, and `geoai`. His projects are available at <https://github.com/pengeos>.

What You Will Learn

- Download and prepare remote sensing data from NAIP, Sentinel-2, and STAC catalogs
- Train deep learning models for segmentation, object detection, and land cover classification
- Apply foundation models like SAM 3, Moondream, and satellite embeddings to real-world problems
- Build interactive maps and visualizations for geospatial analysis
- Use QGIS plugins for segment anything, training, inference, and vision-language models



Scan the QR code to visit the book website and download code examples from <https://book.opengeoai.org>

